



technology services group

Midwest EMC CMA User Group

Upgrading to Documentum 6.5

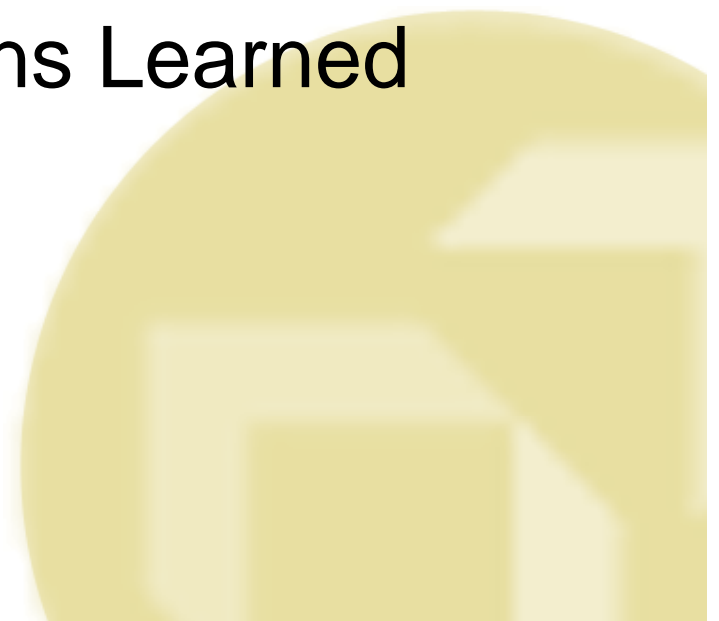
Approaches and Best Practices

March 31, 2009





- Review of D6.5 Features
- Making the Upgrade Decision
- Upgrade Approaches
- Client Case Studies
- Best Practices and Lessons Learned





D6 Release Themes

- Increase Support for Managing High Volume\Large Archives
- Enhanced XML Storage Capabilities
- Start to establish a platform to build Web 2.0-enabled solutions
- Continue to improve user experience in existing applications





General Foundation Updates

- The API is now entirely in Java
 - No more C++ dmcl libraries, dmcl.ini
 - Increases the memory requirements
- JBoss is the embedded application server
 - Tomcat (5.3) -> Weblogic (6.0) -> JBoss (6.5)
 - Simplified administration, smaller footprint
- DFC Updates
 - Dfc.properties is now the only config file
 - No longer a java-com bridge\PIA (.NET impact)
 - Applications need a Global Repository
 - High Volume Server (Scoping, Batch Processing, LWSOs)
 - Aspects available to all as development option



General Foundation Updates

- XML Content Store
 - Integration of the XHive acquisition
 - Implemented as a new type of content store
 - Allows for more native XML management (XQuery, etc.)
- Branch Office Caching Services
 - New model for supporting distributed users
 - 6.5 support for editing at BOCS sites
- Documentum Foundation Services (DFS)
 - Continue to add functionality to the service model
 - Direction is pointing to use DFS vs. DFC



High Volume Server

- Additional functionality to support high volume environments and archive situations
- Several Components:
 - Lightweight System Objects (LwSO) allows objects to share metadata, reducing data footprint and increasing performance (new object model)
 - Batch Processing provides means to perform bulk loading and bulk metadata updates
 - Scoping – allows limits on checks DFC makes on objects
 - Data Partitioning – increased ability to determine “what goes where” in the database
- Intended for static data – if it is “living” content the benefits get taken away pretty quickly
- Requires a separate license



Other Points to Note

- Upgrade Paths
 - Support for upgrade from 5.3 repositories
- HW/SW Requirements
 - No major changes from 6.0 -> 6.5
 - More horsepower and RAM required between 5.3 -> 6.x though, check your current environment
 - Continued support for all the major 3rd party software (OS, app server, RDBMS)
- Additional UI
 - TaskSpace



Webtop 6.5 – Includes 6.0 Additions

- General UI Enhancements
 - Right Click Context Menus, Expandable Columns and Frames
 - Checkboxes Removed
 - Double Click Support
 - Keyboard Shortcuts, Auto Complete
- Multi-Object Handling for Properties
- Streamline Interface Removed
- Presets (configure common settings based on roles)



- Modal Dialogs
- Minimize Refresh
 - Reduces full page refreshes after completing specific actions
- Content Transfer Progress Bar
 - Can display content transfer progress when uploading or viewing content (in UCF or HTTP mode)
- Increased UCF File Transfer Performance
- Multi-Select Drag and Drop
- Deep Folder Export
- Preferences stored in Global Repository
 - no cookies so no longer machine based but user based



Making the Upgrade Decision

To be or not to be, that is the question;

- *William Shakespeare, Hamlet*





Common Upgrade Decision Points

- Support from EMC
- Existing applications and architecture
- Existing integrations
- Business interruption
- Cost
- IT priorities
 - People
 - Hardware/software





Why Upgrade to 6.5?

- Take advantage of new 6.5 features
- Support from EMC:

Platform	End of Primary Support	End of Service Life
5.3 SP5 or earlier	12/31/08	12/31/10
5.3 SP6	12/31/09	12/31/11



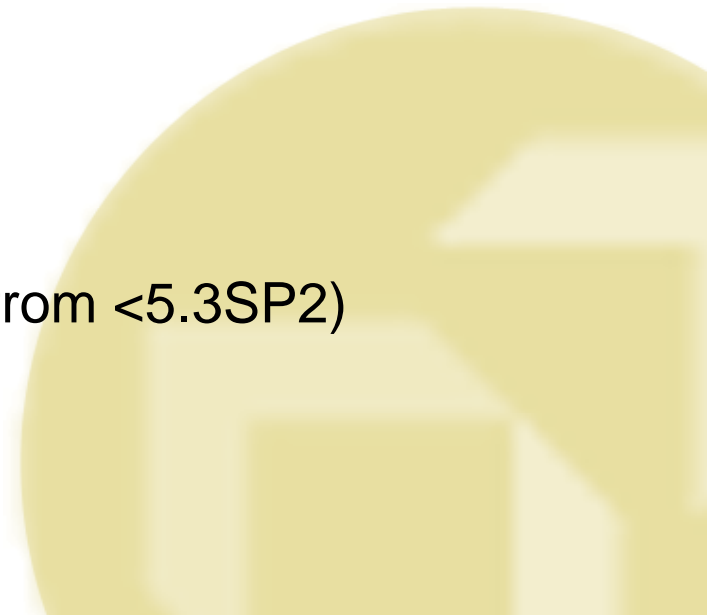
Existing Applications and Architecture

- How many and what kind of applications will impact the scope
 - Custom DFC
 - Custom WDK
 - Configuration and Extension of Webtop
 - .NET
- Current architecture considerations
 - Are those servers big enough?
 - Content Server has 2GB recommended minimum (increase from 5.3)
 - Index Server has 4GB recommended minimum (increase from 5.3)
 - WDK apps require 2GB recommended minimum for each WDK installation on the application server
 - Will moving to 6.5 require other core software updates?
 - WDK apps require Java 1.5 on the application server
 - Other dependencies (RDBMS, OS - Verify with Release Notes)
- If your applications depend on integrations from other vendors, make sure they can support 6.5
 - Annotation Tools (e.g., AnnoDoc)
 - Watermarking (e.g., PDFAqua)



Business Interruption

- System unavailable during upgrade period
 - Downtime will depend on scope of the upgrade
- Level of involvement required from Business Users throughout the process
 - Revisions to system documentation (SOPs, user guides)
 - Re-validation efforts
 - Training
- Other potential impacts
 - Completing running workflows
 - Rebuilding fulltext indexes (coming from <5.3SP2)





The Decision

- After evaluating these factors and coming up with best estimates of total cost, does the benefit outweigh the cost?





Upgrade Approaches - Server

- Content Server Upgrade
 - In place upgrade of all components (Database, Operating System, Content Server)
 - Support for upgrade from 5.3 repositories
 - Need to get to 5.3 first
 - Install D6.5 on new hardware with latest OS, DB, etc. and migrate content into new D6 repository
- Fulltext Index Server
 - In place upgrade of all components while preserving existing index (5.3 SP2 or higher)
 - Need to rebuild the index if coming from <5.3SP2



Upgrade Approaches - Applications

- Custom DFC Applications
 - Update deployment package to include DFC 6 jar files (No longer depends on DMCL installation)
 - Java 1.5 Required by Newer Version of Apache Tomcat
- WDK Applications
 - Custom WDK folder can be migrated to D6
 - However new 6.5 features and/or changes may drive necessary updates to customizations
 - WDK configuration or class updates\changes
 - D5 action pre-conditions vs. D6 presets
 - No streamline view in D6
- .NET Applications
 - PIA no longer available in 6.5
 - Update to use DFS or other “creative” solution





Client Case Study 1

- **Who** – Pharmaceutical Manufacturer
- **What** – Upgrade of one application\repository from customized Webtop 5.3\Content Server 5.3 to Webtop 6.5\Content Server 6.5
- **When** – 2008 - 2009
- **System Details** –
 - Currently supports multiple sites in the U.S.
 - Manages quality documentation (SOPs, etc.)
- **Why** –
 - Expanding rollout to international sites
 - Business required enhancements to the application
 - Move to a supported version, 6.5 SP1





Project Objectives

- Deliver required enhancements to the business users
 - Lifecycle enhancements
 - Reporting enhancements
- Rollout the application to international sites
- Move to the latest supported version

Platform	End of Primary Support	End of Service Life
5.3 SP5 or earlier	12/31/08	12/31/10
5.3 SP6	12/31/09	12/31/11
D6	8/31/10	08/31/12



Content Server Approach

- In Place Upgrade
 - Currently on 5.3 SP4
 - Server specifications still enough to meet the requirements
- Test against production data
 - Created a complete clone environment
 - Needed to procure temporary servers
 - Also leveraged virtual environments
 - Exported copy of production data
 - Executed the 6.5 upgrade
- Index Server
 - Upgrade index server components
 - No need to rebuild indexes
- Determine when ready for production upgrade
 - Clean upgrades in test environment
 - Build upgrade task list





Application Approach

- Currently using customized Webtop 5.3
 - ~35 extended components
 - Recompile extended classes with 6.5
 - Provides immediate feedback on what is broken
 - Resolve exceptions
 - Deploy custom folder in 6.5
 - Make sure xml configs are still functional
- Develop new enhancements in 6.5
 - Take advantage of 6.5 functionality and UI
 - Implement some functionality as pre-sets



Migrate Webtop Customizations

- Compare XML Configurations – D5.3

The screenshot shows a Notepad++ window with the following XML content:

```
27 <component id="checkin">
28   <ucfrequired/>
29   <params>
30     <param name="objectId" required="true"/>
31     <param name="vdmRootObjectId" required="false"/>
32     <param name="nodeId" required="false"/>
33   </params>
```

The status bar at the bottom indicates: eXtensible Markup nb char : 3299 | Ln : 1 Col : 1 Sel : 0 | Dos\Windows ANSI | INS



Migrate Webtop Customizations

- Compare XML Configurations – D6

The screenshot shows a Notepad++ window with the title "Notepad++ - C:\Documents and Settings\gsunko\Desktop\checkin_component_D...". The menu bar includes File, Edit, Search, View, Format, Language, Settings, Macro, Run, TextFX, Plugins, Window, and ?. The toolbar contains various icons for file operations and editing. The active tab is "checkin_component_D6.xml". The main text area displays the following XML code:

```
27 <component id="checkin">
28   <params>
29     <param name="objectId" required="true" />
30     <param name="vdmRootObjectId" required="false" />
31     <param name="nodeId" required="false" />
32     <param name="silent" required="false" />
33     <param name="version" required="false" />
34     <param name="retainLock" required="false" />
35     <param name="keepLocalFile" required="false" />
36   </params>
```

The status bar at the bottom shows "eXtensible Markup L nb char : 5848", "Ln : 28 Col : 1 Sel : 0", "Dos\Windows", "ANSI", and "INS".



Webtop Customization

- Updated any configuration node overwritten by your customizations
- Verified no changes to Java methods or JSP pages being overridden
 - Menubar.jsp
 - Search packages
 - Check changes documentation from EMC
- Test, test, test





Lessons Learned

- In place upgrade
 - Steps need to be followed and documented – build a checklist!
 - Testing against a production clone is essential
- Application Upgrade
 - Do not underestimate potential work of checking through your WDK extensions
 - Having good documentation to test against (test scripts, requirements) will help make sure that even when everything compiles and starts the functionality is still there
- Business Impact
 - Do not underestimate changes to UI even though they may seem subtle compared with 5.3
 - Training estimate had to be increased before rollout



Client Case Study 2

- **Who** - Medical device industry client: R&D and manufacturing
- **What** - Research project to upgrade and consolidate a 5.2.5 Content Server environment to Documentum 6.5
- **When** – 2009 and 2010
- **Where** – Global system managed in the US
 - Controlled documentation
 - Manufacturing documentation archival
 - General quality documentation for products
- **Why** –
 - Move to a supported version, 6.5 SP1
 - Use new Documentum and custom client apps
 - Evaluate High Volume Server
 - Consolidate existing Windows and Unix platforms
 - Consolidate multiple archive Docbases



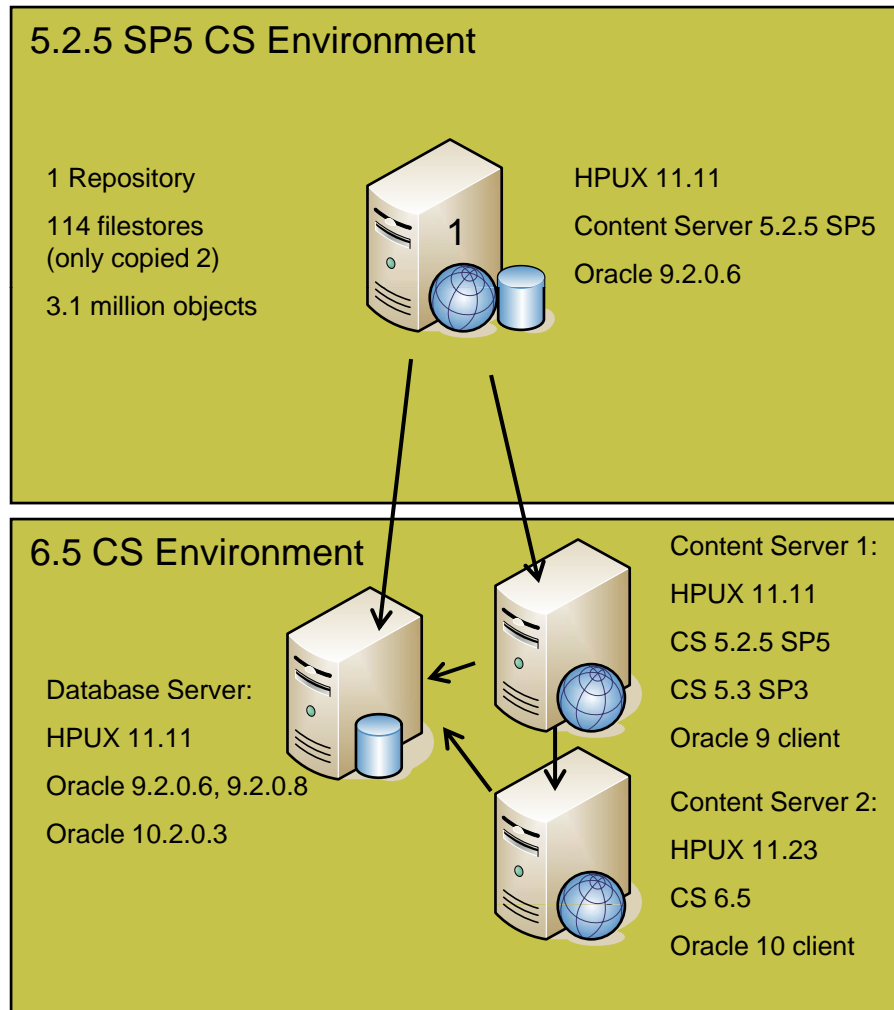
Project Objectives

- Find the “best” way to upgrade Documentum systems
 - Clone/In Place
 - Migrate
- Determine if client applications will continue to function
- Evaluate new 6.5 High Volume Server functions
 - LWSOs
 - Partitioning

Platform	End of Primary Support	End of Service Life
5.3 SP5 or earlier	12/31/08	12/31/10
5.3 SP6	12/31/09	12/31/11
D6	8/31/10	08/31/12



Clone / In Place Upgrade



High Level Steps

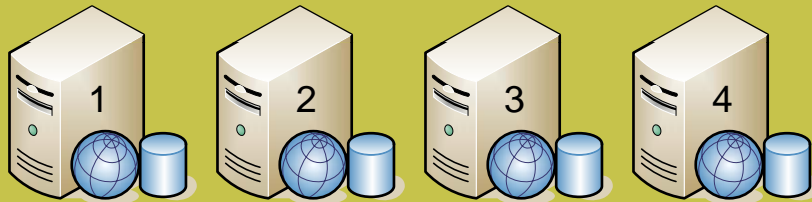
1. Set up Oracle 9.2.0.6 on target database server, HPUX 11.11
2. Install 5.2.5 SP5 Content Server on HPUX 11.11
3. Export/Import Oracle data and filestore 01 and 04 data to create a clone
4. Test the clone and backup
5. Upgrade Oracle to 9.2.0.8
6. Install Content Server 5.3 SP6 and upgrade the Repository
7. Clone 5.3 SP6 Content Server install to HPUX 11.23 server
 - Couldn't upgrade OS in place
8. Install Oracle 10.2.0.3 on database server
9. Perform Unicode conversion on the database and import to Oracle 10
10. Install Content Server to 6.5 and upgrade the Repository



Migration Upgrade

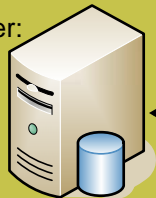
5.2.5 SP5 CS Environment

HPUX 11.11	Win 2K	Win 2K	Win 2K
Oracle 9.2.0.6	Oracle 8.1.6	Oracle 8.1.6	Oracle 8.1.7
CS 5.2.5 SP5	CS 5.2.5	CS 5.2.5	CS 5.2.5



6.5 CS Environment

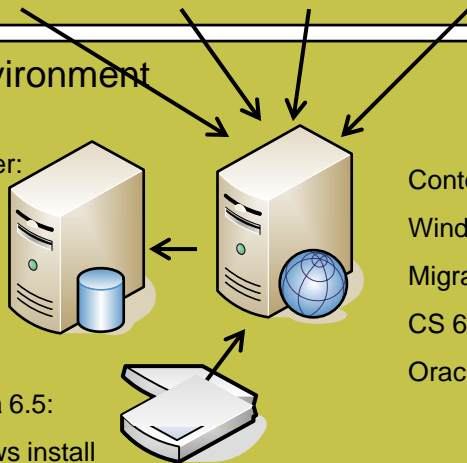
Database Server:
HPUX 11.11
Oracle 10.2.0.3



Content Server:
Windows 2003
Migration Utility
CS 6.5
Oracle 10 client



Captiva 6.5:
Windows install
2,700 docs/day



# of	Docs (millions)	Groups	ACLs	Custom Types	Attrs	Cabinets / Folders	Named Users
1	2.1	5	5	11	161	9	38
2	1.8	5	5	7	25	9	38
3	0.5	5	5	7	25	9	38
4	0.1	160	117	0	0	35	723

High Level Steps

1. Use Oracle 10.2.0.3 install from In Place upgrade – set up new tablespace
2. Install CS 6.5 on Windows 2003 server and create new Docbroker and Repository
3. Configure new Repository to accept content from the four source Repositories: DQL, API, and User
4. Run test migrations for each document type from each source Repository

Notes

1. Consolidation for repositories 2 & 3.
2. Repositories 1 & 4 were migrated to a single repository for testing only
3. A full migration was not done on the 5.2.5 repositories
4. Repositories 1, 2, and 3 are licensed for anonymous users
5. Repository 1 also has custom jobs, procedures, and methods



Client Applications

Does it Work?	6.5 Administrator and Webtop	.NET Web App (using PIA)	Migration Tool	VB 6.0/ASP DMCL API	Captiva 6.5
5.2.5 SP5 CS	No	Yes	Yes	Yes	Not tested
5.3 SP6 CS	Yes	Yes	Yes	Yes	Not tested
6.5 CS	Yes	Yes	Yes	Yes	Yes
DFC 5.3 SP6	Not tested	Yes	Yes	Yes	Not tested
DFC 6.5	Yes	No	Yes	No	Yes

- The 5.3 SP6 DFC was used to facilitate connections to the 5.2.5 SP5 Docbase and the 6.5 Docbase
- .NET PIA is shipped with 5.3 SP6 and 6.0, but not with 6.5
 - .NET PIA does not have any 6.x functionality
 - LWSO would need to be created using 6.5 Administration Method
- DMCL API file used is dmclvb40.dll
 - Ships with DFC 5.3 SP6 – does not ship with 6.5



- Repository cannot continue to use “support” as an object type name
- LWSOs do not allow for versioning
 - Require business and client application changes to use
- Partitioning requires additional DBA involvement and maintenance
 - Low volume of scanned documents; <3,000 per day
- .NET support really is gone; not available for LWSOs
- Lots of new HW needed; eventually we can reclaim the old HW
- The Unicode conversion of the database took several days. Questions remain about how fast it can be accomplished



Lessons Learned

- In place upgrade
 - takes a long time
 - Heavy DBA involvement
 - Heavy UNIX server admin involvement
 - Steps needed to be very specific and well coordinated
- Migration
 - Lots of scripting
 - Requires a rigorous data checking process
- DMCL emulator is used by the Content Server, not needed for the client applications
- WorkSpace does not work with 6.5 CS
- Could use DFC 5.3 SP6 version for client apps
 - except for LWSOs, that needs 6.x DFC



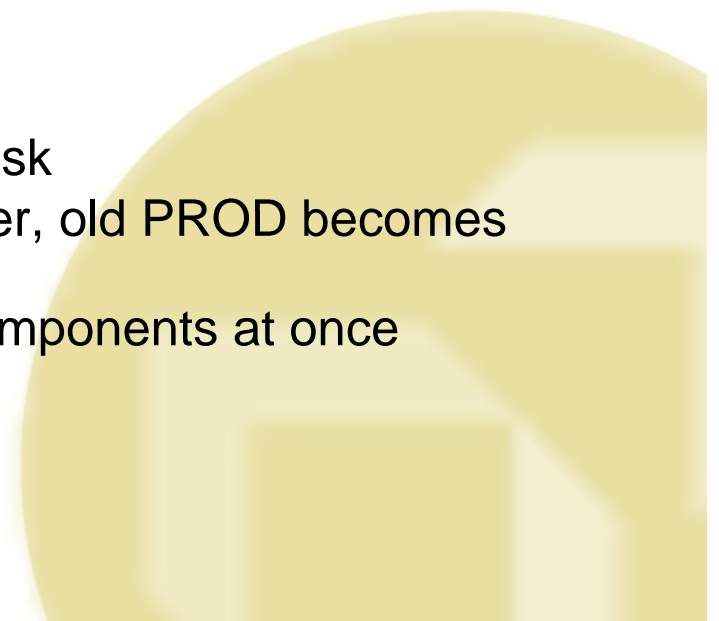
- Opted for no Fulltext Indexing
 - Reduces HW
 - Reduce complexity
- Migrate to save DBA and Unix Admin effort
 - Resources are already strained
- Migrate to keep heaviest use of resources in the application team
- Jury is still out on value of LWSOs and partitioning
 - Requires application changes
 - Requires business process changes
 - Heavy DBA resource requirements





Best Practices for Managing the Upgrade Process

- Develop applications/object model with upgrades in mind
 - WDK/Webtop/DAMtop/Web Publisher will change
 - Customizations should be able to upgrade easily between versions
 - Isolate technology layers
- Utilize new hardware for significant release
 - Testing is more straightforward
 - Mock migration runs are simple
 - Fall back is zero-effort
 - Cost nearly always outweighed by the risk
 - TEST becomes new PROD; post-cutover, old PROD becomes new TEST
 - Reduces risk of upgrades to multiple components at once (RDBMS, OS, etc.)





Best Practices for Managing the Upgrade Process

- Existing Documentum users may want to wait until platform is stable
 - X.0
 - X.1
 - X.1 SP1 – Production
- Business benefits need to justify cost and effort
 - Weigh new interface features versus additional training
 - Old customizations versus rebuilding on new platform





Best Practices

- Make sure your environment is in good shape for completing the upgrade
 - Review current capacity vs requirements and future needs
 - Review Content Server logs and trace files for discrepancies
 - Review relevant jobs for status on repositories
- Better to resolve issues during planning and test upgrades than during the production run!



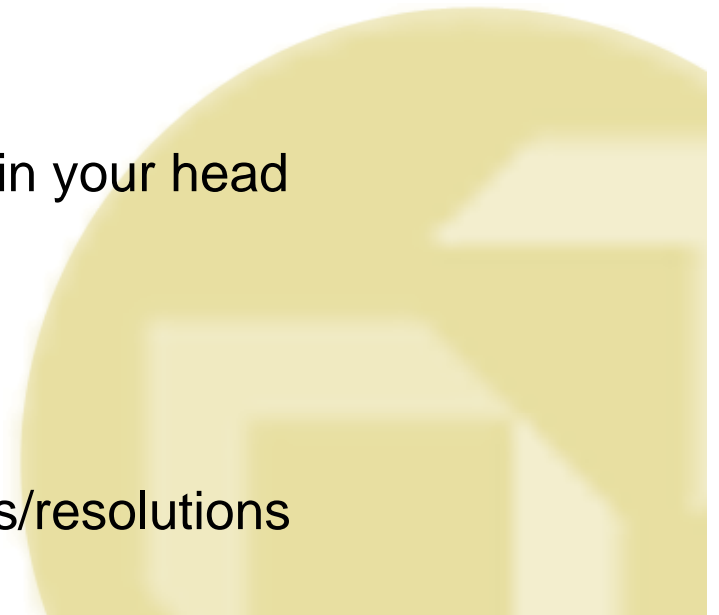
Upgrade Planning

- Identify software/hardware requirements
- Reviewing sizing guide
- Identify IT personnel
 - Mark time on calendars
 - Make sure all parties (DBA, etc.) are aware of what needs to be done, access requirements, etc.
- Develop upgrade task list based on install guides and release notes
- Develop initial qualification scripts
- Plan backup and rollback strategy



Test the Upgrade

- If at all possible, test on as exact a clone of production as possible
 - Allows for as close a test as possible
 - Also useful for developing metrics
- Follow developed upgrade checklist
 - Note any discrepancies or exceptions and how they were resolved
 - Capture upgrade screenshots
 - Adds time but do not just let it remain in your head
- Capture upgrade screenshots
- Run qualification scripts
 - Note any errors or exceptions
 - Come up with appropriate adjustments/resolutions





Production Upgrade

- Schedule the upgrade window based on collected metrics
- Make sure all resources are aware and available
 - Potential for RDBMS access
 - Potential for Server access
- Perform a last check of the repository status as close to performing the upgrade as possible to prevent last minute complications
- Follow the upgrade checklist based off of the testing performed against the production clone



technology services group

Questions/Follow Up Contact Information

- Jonathan Paetsch
 - 312-372-7777 x262
 - jonathanp@tsgrp.com

